AMENDMENTS TO THE SPECIFICATION

Please replace paragraph [0021] with the following amended paragraph:

[0021] The present invention extends to methods, systems and computer program products for authorizing a requesting entity in a manner that is at least partially independent of the underlying target data structure that is desired to be accessed. In one operating environment, there are a number of individuals and applications operating through a variety of services on a variety of different types of identity-specific data structures that are organized in accordance with a set [[or]] of rules. Each service is configured to perform operations on one or more different types of data structures. For example, an identity may have an in-box data structure organized in accordance with an in-box schema and that is managed by an in-box service, a calendar data structure organized in accordance with a calendar schema and that is managed by a calendar service, and so forth.

Please replace paragraph [0029] with the following amended paragraph:

[0029] This operating network is just one example of many possible network operating environments in which the present invention may be employed. For example, as represented by the ellipses in Figure 1, the number of applications 110 and services 120 that interact with the authorization station 130 may vary. In addition, each service may manage more than just one different type of data structure. Furthermore, there is no requirement that the data being managed by the service be identity-specific although that is one specific implementation. Also, many of the services may employ additional authorization functionality not performed by the authorization station 130. Furthermore, although only one centralized authorization station is shown and described for clarity, there may be more than one (and even numerous) authorization stations that perform the described authorization on behalf of the services. And finally, some of the functionality of the authorization station 130 may also be employed instead by the services.

Please replace paragraph [0042] with the following amended paragraph:

[0042] Figure 4 illustrates a method 400 for authorizing a requesting entity to operate upon a data structure in a standard manner that is at least partially, if not wholly, independent of the type of underlying data structure being operated upon. The method 400 first includes an act of maintaining a number of role templates (act 401) that each define basic access permissions with

respect to a number of command methods, wherein at least some of the role templates define access permissions in a manner that is independent of the type of data structure being accessed. In Figure 2, these role templates may be maintained in, for example, the role list documents 227 at the XML store 225.

Please replace paragraph [0069] with the following amended paragraph:

[0069] In summary, the role map defines general access permissions using role templates that refer to scopes. In one embodiment, role templates are generated that are representative of typical access permissions that might be desired by users using the particular service. Referring to Figure 4, the authorization station maintains these role templates and thus accomplishes the act of maintaining a number of role templates (act 401) that each define basic access permissions with respect to a number of command methods, wherein at least some of the role templates define access permissions in a manner that is independent of the type of data structure being accessed. Note that the target service is only factored in when selecting a role map to use. The process then remains the same regardless of the target service.

Please replace paragraph [0073] with the following amended paragraph:

[0073] The "role list" contains a number of scopes that allow for more fine-grained defining of scopes. The scopes may follow the same scope schema that was described further above. The role list also [[include]] includes any number of role definitions in the form of "role" elements. The various elements and attributes will now be described.

Please replace paragraph [0074] with the following amended paragraph:

[0074] the "roleList/@changeNumber" attribute facilitates caching of the element and its descendants. The "roleList/@instanceId" attribute is a unique identifier typically assigned to the root element of a service. The "rolelist/role" is a role definition that matches a particular requesting entity with particular access rights. These access rights are defined by any applicable scope referred to in any applicable role template, along with any scope directly referred to in the role definition itself.

Please replace paragraph [0075] with the following amended paragraph:

[0075] The "roleList/role/@scopeRef" attribute specifies the scope in the role list that is in effect for this role definition. The "roleList/role/@roleTemplateRef" attribute specifies the name of the role template in the service's role map that this role definition is bound to. The "roleList/role/@changeNumber" attribute is designed to facilitate caching of the element and its descendants. The "/roleList/role/@id" attribute is a globally unique ID assigned to this element. The "roleList/role/@creator" attribute identifies the creator in terms of user identifier, application identifier, and platform identifier. The "roleList/role/cat" categorizes the element that contains it by referencing a global category. The "roleList/role/notes" specifies notes that may be used to specify reasoning being adding this role to the roleList.

Please replace paragraph [0084] with the following amended paragraph:

[0084] Then, a third matching operation is performed in which the combined platform identifier and application identifier of the request are matched against the "appAndPlatformId" attribute of the second set of role definitions. This generates a third set of role definitions. If there are no role definitions in the third set of role definitions, then all role definitions from the second set having a "subject" elements containing the "appAndPlatformId" attribute are discarded keeping only those "subject" elements that do not contain an "appAndPlatformId" attribute to form the third set of role definitions. If a matching role element is not found, the request is failed with an authorization fault. Also, if the matching role definition contains an "expiresAt/" element that [[indicate]] indicates that the role definition has expired, then an error message is also returned.